

Claims Amendment for Enabling Examiner's Amendment

1. (Currently Amended) A computer implemented method for producing data domain information structures and associated values, sets of data values and/or tuples for testing or validation of a computer program, to be used for data structure elements of a computer program during testing and validation of the computer program, the method comprising:

generating ~~receiving~~ a reflection of the computer program, ~~the computer program~~ comprising constructs, wherein the computer program constructs comprise one or more first data structure elements and a second data structure element, the one or more first data structure elements and the second data structure element comprising data types, data fields, functions, methods and/or parameters of the computer program;

annotating the one or more first data structure elements and the second data structure element with information related to the purpose of domain configuration;

configuring one or more first data domains corresponding to the one or more first data structure elements and a second data domain corresponding to the second data structure element, the configuring comprising generating one or more structural language expressions for representing/evaluating the one or more first data structure elements and the second data structure element and for defining relationships among the one or more first data structure elements and the second data structure element in terms of type/method inheritance and types;

wherein generating the one or more structural language expressions includes generating at least one of:

if the second data structure element is a first data type comprising a plurality a sub-types, at least one structural language expression denoting a union of one or more of the one or more first data domains corresponding to the one or more first data structure elements that are sub-types of the first data type;

if the second data structure element is a sub-type, data field or parameter of a second data type, at least one structural language expression indicating that the second data domain can be inherited from one of the one or more first data structure elements that are of the second data type;

if the second data structure element is a method, at least one structural language expression denoting inheritance relationships between types and subtypes of the second data structure element and types and subtypes of the one or more first data structure

elements wherein the one or more first data structure elements represent parameters of another method described in the one or more first data domains; and

at least one structural language expression comprising a Cartesian product as an invocable method to cross-relate the second data structure element with one or more data domains corresponding to the one or more first data structure elements; and

compiling the annotated one or more first data structure elements and the second data structure element to produce values, sets of values and/or tuples of the one or more first data domains and the second data domain; and

testing or validating the computer program using at least one of the values, sets of values and/or tuples from the one or more first data domains and the second data domain,

receiving domain-configuration information comprising:

—— one or more first mathematical or programmatic expressions related to configuring one or more first data domains corresponding to the one or more first data structure elements, each of the one or more first data domains corresponding to one of the one or more first data structure elements; and

—— one or more second mathematical or programmatic expressions related to configuring a second data domain corresponding to the second data structure element, the second mathematical or programmatic expressions operable on the one or more first data domains;

producing the one or more first data domains by evaluating the one or more first mathematical or programmatic expressions, the one or more first data domains comprising one or more values or tuples generated from evaluating the one or more first mathematical or programmatic expressions; and

producing the second data domain by evaluating the one or more second mathematical or programmatic expressions, the second data domain comprising one or more values or tuples generated from evaluating the one or more second mathematical or programmatic expressions;

and

wherein each data domain defines a set of values or tuples to be used for the corresponding data structure element during testing and validation of the computer program, and the second data domain contains at least one value or tuple not included in the one or more first data domains;

2-7. (Canceled)

8. (Currently Amended) The method of claim 1, wherein the one or more ~~first mathematical or programmatic expressions or the one or more second mathematical or programmatic~~ structural language expressions comprise methods and functions defined within the code of the computer program, which are exposed via the reflection of the computer program.

9-15. (Canceled)

16. (Currently Amended) The method of claim 1, wherein the one or more structural language ~~second mathematical or programmatic~~ expressions comprise[[s]] one or more predicates such that the second data domain comprises values or tuples that meet the one or more predicate.

17-18. (Canceled)

19. (Currently Amended) One or more computer-readable media storing computer-executable instructions for performing a method for producing data domain information structures and associated values, sets of [[data]] values and/or tuples ~~to be used for data structure elements of a computer program during~~ for testing [[and]] or validation of [[the]] a computer program, the method comprising:

generating a reflection of the computer program comprising constructs, wherein the computer program constructs comprise one or more first data structure elements and a second data structure element, the one or more first data structure elements and the second data structure element comprising data types, data fields, functions, methods and/or parameters of the computer program;

annotating the one or more first data structure elements and the second data structure element with information related to the purpose of domain configuration;

configuring one or more first data domains corresponding to the one or more first data structure elements and a second data domain corresponding to the second data structure element, the configuring comprising generating one or more structural language expressions for representing/evaluating the one or more first data structure elements and the second data structure element and for defining relationships among the one or more first data structure elements and the second data structure element in terms of type/method inheritance and types;

wherein generating the one or more structural language expressions includes generating at least one of:

if the second data structure element is a first data type comprising a plurality of sub-types, at least one structural language expression denoting a union of one or more of the one or more first data domains corresponding to the one or more first data structure elements that are sub-types of the first data type;

if the second data structure element is a sub-type, data field or parameter of a second data type, at least one structural language expression indicating that the second data domain can be inherited from one of the one or more first data structure elements that are of the second data type;

if the second data structure element is a method, at least one structural language expression denoting inheritance relationships between types and subtypes of the second data structure element and types and subtypes of the one or more first data structure elements wherein the one or more first data structure elements represent parameters of another method described in the one or more first data domains; and

at least one structural language expression comprising a Cartesian product as an invocable method to cross-relate the second data structure element with one or more data domains corresponding to the one or more first data structure elements; and
compiling the annotated one or more first data structure elements and the second data structure element to produce values, sets of values and/or tuples of the one or more first data domains and the second data domain; and

testing or validating the computer program using at least one of the values, sets of values and/or tuples from the one or more first data domains and the second data domain.

24. (Currently Amended) The one or more computer-readable media of claim 19, wherein the one or more ~~first mathematical or programmatic expressions or the one or more second mathematical or programmatic~~ structural language expressions comprise methods and functions defined within the computer program and exposed via the reflection of the computer program.

25-35. (Canceled)

36. (Currently Amended) The method of claim 1, wherein the one or more ~~first and one one or more second mathematical or programmatic~~ structural language expressions are provided ~~in~~ as Abstract State Machine Language (ASML) expressions.

37-39. (Canceled)

40. (Currently Amended) The one or more computer-readable media of claim 19, wherein the one or more ~~second mathematical or programmatic~~ structural language expressions comprise[[s]] one or more predicates such that the second data domain comprises values or tuples that meet the one or more predicates.

41-42. (Canceled)

Included is a Claims Amendment for Enabling Examiner's Amendment in the above-referenced matter. The attorney of record, Stephen A. Wight (Registration No. 37,759), has authorized the amendment.

Should you have any question regarding this amendment, I will be available by email Saturday and Sunday.

Regards,

Doug Dallmann

Patent Attorney (Reg. No. 65,636)
Klarquist Sparkman, LLP
Intellectual Property Law
One World Trade Center, Suite 1600
Portland, OR 97204-2988 U.S.A.
(503) 595-5300 | (503) 595-5301 fax